Working With Missing Data in Pandas

September 29, 2025

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
d = {'First Score': [100, 90, np.nan, 95],
     'Second Score': [30, 45, 56, np.nan],
     'Third Score': [np.nan, 40, 80, 98]}
df = pd.DataFrame(d)
```

[4]:
```
df
```

[4]:

	First Score	Second Score	Third Score
0	100.0	30.0	NaN
1	90.0	45.0	40.0
2	NaN	56.0	80.0
3	95.0	NaN	98.0

[5]:
```
mv = df.isnull()
```

[6]:
```
print(mv)
```

```
   First Score  Second Score  Third Score
0        False         False         True
1        False         False        False
2         True         False        False
3        False          True        False
```

[13]:
```
d = pd.read_csv(r"content\employees.csv")
```

[14]:
```
bool_series = pd.isnull(d["Gender"])
missing_gender_data = d[bool_series]
print(missing_gender_data)
```

```
    First Name Gender  Start Date Last Login Time  Salary  Bonus %  \
20        Lois    NaN   4/22/1995         7:18 PM   64714    4.934
22      Joshua    NaN    3/8/2012         1:58 AM   90816   18.816
27       Scott    NaN   7/11/1991         6:58 PM  122367    5.218
31       Joyce    NaN   2/20/2005         2:40 PM   88657   12.752
41   Christine    NaN   6/28/2015         1:08 AM   66582   11.308
..         ...    ...         ...             ...     ...      ...
```

```
961    Antonio    NaN   6/18/1989         9:37 PM  103050    3.050
972     Victor    NaN   7/28/2006         2:49 PM   76381   11.159
985    Stephen    NaN   7/10/1983         8:10 PM   85668    1.909
989     Justin    NaN   2/10/1991         4:58 PM   38344    3.794
995      Henry    NaN  11/23/2014         6:09 AM  132483   16.655

    Senior Management                 Team
20               True                Legal
22               True       Client Services
27              False                Legal
31              False              Product
41               True  Business Development
..                …                    …
961             False                Legal
972              True                Sales
985             False                Legal
989             False                Legal
995             False          Distribution

[145 rows x 8 columns]
```

[15]:
```
data = {'Name': ['Amit', 'Sita', np.nan, 'Raj'],
        'Age': [25, np.nan, 22, 28]}

df = pd.DataFrame(data)

# Check for missing values using isna()
print(df.isna())
```

```
    Name    Age
0  False  False
1  False   True
2   True  False
3  False  False
```

[16]:
```
d = {'First Score': [100, 90, np.nan, 95],
        'Second Score': [30, 45, 56, np.nan],
        'Third Score': [np.nan, 40, 80, 98]}
df = pd.DataFrame(d)

nmv = df.notnull()

print(nmv)
```

```
   First Score  Second Score  Third Score
0         True          True        False
1         True          True         True
2        False          True         True
3         True         False         True
```

[18]:
```
d = pd.read_csv(r"content\employees.csv")

nmg = pd.notnull(d["Gender"])

nmgd= d[nmg]

display(nmgd)
```

```
    First Name  Gender Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male   8/6/1993        12:42 PM   97308    6.945
1       Thomas    Male  3/31/1996         6:53 AM   61933    4.170
2        Maria  Female  4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male   3/4/2005         1:00 PM  138705    9.340
4        Larry    Male  1/24/1998         4:47 PM  101004    1.389
..         …     …        …               …     …      …
994     George    Male  6/21/2013         5:47 PM   98874    4.479
996    Phillip    Male  1/31/1984         6:30 AM   42392   19.675
997    Russell    Male  5/20/2013        12:39 PM   96914    1.421
998      Larry    Male  4/20/2013         4:45 PM   60500   11.985
999     Albert    Male  5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
1                True                  NaN
2               False              Finance
3                True              Finance
4                True       Client Services
..                …                    …
994              True            Marketing
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[855 rows x 8 columns]
```

[19]:
```
d = {'First Score': [100, 90, np.nan, 95],
        'Second Score': [30, 45, 56, np.nan],
        'Third Score': [np.nan, 40, 80, 98]}
df = pd.DataFrame(d)

df.fillna(0)
```

[19]:

	First Score	Second Score	Third Score
0	100.0	30.0	0.0
1	90.0	45.0	40.0
2	0.0	56.0	80.0
3	95.0	0.0	98.0

[21]:
```
df.fillna(method='pad') # Fill with previous value (forward fill)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_19020\1069722036.py:1: FutureWarning:
DataFrame.fillna with 'method' is deprecated and will raise in a future version.
Use obj.ffill() or obj.bfill() instead.
  df.fillna(method='pad') # Fill with previous value (forward fill)
```

[21]:

	First Score	Second Score	Third Score
0	100.0	30.0	NaN
1	90.0	45.0	40.0
2	90.0	56.0	80.0
3	95.0	56.0	98.0

[22]:
```
df.fillna(method='bfill') # Fill with next value (backward fill)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_19020\202629142.py:1: FutureWarning:
DataFrame.fillna with 'method' is deprecated and will raise in a future version.
Use obj.ffill() or obj.bfill() instead.
  df.fillna(method='bfill') # Fill with next value (backward fill)
```

[22]:

	First Score	Second Score	Third Score
0	100.0	30.0	40.0
1	90.0	45.0	40.0
2	95.0	56.0	80.0
3	95.0	NaN	98.0

[25]:
```
d = pd.read_csv(r"content\employees.csv")

d[10:25]
```

[25]:

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
10	Louise	Female	8/12/1980	9:01 AM	63241	15.132	
11	Julie	Female	10/26/1997	3:19 PM	102508	12.637	
12	Brandon	Male	12/1/1980	1:08 AM	112807	17.492	
13	Gary	Male	1/27/2008	11:40 PM	109831	5.831	
14	Kimberly	Female	1/14/1999	7:13 AM	41426	14.543	
15	Lillian	Female	6/5/2016	6:09 AM	59414	1.256	
16	Jeremy	Male	9/21/2010	5:56 AM	90370	7.369	
17	Shawn	Male	12/7/1986	7:45 PM	111737	6.414	
18	Diana	Female	10/23/1981	10:27 AM	132940	19.082	
19	Donna	Female	7/22/2010	3:48 AM	81014	1.894	
20	Lois	NaN	4/22/1995	7:18 PM	64714	4.934	
21	Matthew	Male	9/5/1995	2:12 AM	100612	13.645	
22	Joshua	NaN	3/8/2012	1:58 AM	90816	18.816	
23	NaN	Male	6/14/2012	4:19 PM	125792	5.042	
24	John	Male	7/1/1992	10:08 PM	97950	13.873	

	Senior Management	Team

10	True	NaN	
11	True	Legal	
12	True	Human Resources	
13	False	Sales	
14	True	Finance	
15	False	Product	
16	False	Human Resources	
17	False	Product	
18	False	Client Services	
19	False	Product	
20	True	Legal	
21	False	Marketing	
22	True	Client Services	
23	NaN	NaN	
24	False	Client Services	

[26]:
```
d["Gender"].fillna('No Gender', inplace = True)
d[10:25]
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_19020\1225287812.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.
```

```
d["Gender"].fillna('No Gender', inplace = True)
```

[26]:

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
10	Louise	Female	8/12/1980	9:01 AM	63241	15.132	
11	Julie	Female	10/26/1997	3:19 PM	102508	12.637	
12	Brandon	Male	12/1/1980	1:08 AM	112807	17.492	
13	Gary	Male	1/27/2008	11:40 PM	109831	5.831	
14	Kimberly	Female	1/14/1999	7:13 AM	41426	14.543	
15	Lillian	Female	6/5/2016	6:09 AM	59414	1.256	
16	Jeremy	Male	9/21/2010	5:56 AM	90370	7.369	
17	Shawn	Male	12/7/1986	7:45 PM	111737	6.414	
18	Diana	Female	10/23/1981	10:27 AM	132940	19.082	
19	Donna	Female	7/22/2010	3:48 AM	81014	1.894	
20	Lois	No Gender	4/22/1995	7:18 PM	64714	4.934	
21	Matthew	Male	9/5/1995	2:12 AM	100612	13.645	
22	Joshua	No Gender	3/8/2012	1:58 AM	90816	18.816	
23	NaN	Male	6/14/2012	4:19 PM	125792	5.042	

```
24       John       Male    7/1/1992        10:08 PM   97950   13.873

   Senior Management             Team
10              True              NaN
11              True            Legal
12              True  Human Resources
13             False            Sales
14              True          Finance
15             False          Product
16             False  Human Resources
17             False          Product
18             False  Client Services
19             False          Product
20              True            Legal
21             False        Marketing
22              True  Client Services
23               NaN              NaN
24             False  Client Services
```

[27]:
```
data = pd.read_csv(r"content\employees.csv")
data[10:25]
```

[27]:
```
   First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
10     Louise  Female   8/12/1980         9:01 AM   63241   15.132
11      Julie  Female  10/26/1997         3:19 PM  102508   12.637
12    Brandon    Male   12/1/1980         1:08 AM  112807   17.492
13       Gary    Male   1/27/2008        11:40 PM  109831    5.831
14   Kimberly  Female   1/14/1999         7:13 AM   41426   14.543
15    Lillian  Female    6/5/2016         6:09 AM   59414    1.256
16     Jeremy    Male   9/21/2010         5:56 AM   90370    7.369
17      Shawn    Male   12/7/1986         7:45 PM  111737    6.414
18      Diana  Female  10/23/1981        10:27 AM  132940   19.082
19      Donna  Female   7/22/2010         3:48 AM   81014    1.894
20       Lois     NaN   4/22/1995         7:18 PM   64714    4.934
21    Matthew    Male    9/5/1995         2:12 AM  100612   13.645
22     Joshua     NaN    3/8/2012         1:58 AM   90816   18.816
23        NaN    Male   6/14/2012         4:19 PM  125792    5.042
24       John    Male    7/1/1992        10:08 PM   97950   13.873

   Senior Management             Team
10              True              NaN
11              True            Legal
12              True  Human Resources
13             False            Sales
14              True          Finance
15             False          Product
16             False  Human Resources
```

```
17             False          Product
18             False  Client Services
19             False          Product
20              True            Legal
21             False        Marketing
22              True  Client Services
23               NaN              NaN
24             False  Client Services
```

[31]:
```
data.replace(to_replace=np.nan, value=-99, inplace=True)
data[10:25]
```

[31]:
```
   First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
10     Louise  Female   8/12/1980         9:01 AM   63241   15.132
11      Julie  Female  10/26/1997         3:19 PM  102508   12.637
12    Brandon    Male   12/1/1980         1:08 AM  112807   17.492
13       Gary    Male   1/27/2008        11:40 PM  109831    5.831
14   Kimberly  Female   1/14/1999         7:13 AM   41426   14.543
15    Lillian  Female    6/5/2016         6:09 AM   59414    1.256
16     Jeremy    Male   9/21/2010         5:56 AM   90370    7.369
17      Shawn    Male   12/7/1986         7:45 PM  111737    6.414
18      Diana  Female  10/23/1981        10:27 AM  132940   19.082
19      Donna  Female   7/22/2010         3:48 AM   81014    1.894
20       Lois     -99   4/22/1995         7:18 PM   64714    4.934
21    Matthew    Male    9/5/1995         2:12 AM  100612   13.645
22     Joshua     -99    3/8/2012         1:58 AM   90816   18.816
23        -99    Male   6/14/2012         4:19 PM  125792    5.042
24       John    Male    7/1/1992        10:08 PM   97950   13.873

   Senior Management             Team
10              True              -99
11              True            Legal
12              True  Human Resources
13             False            Sales
14              True          Finance
15             False          Product
16             False  Human Resources
17             False          Product
18             False  Client Services
19             False          Product
20              True            Legal
21             False        Marketing
22              True  Client Services
23               -99              -99
24             False  Client Services
```

[32]:
```
df = pd.DataFrame({"A": [12, 4, 5, None, 1],
                   "B": [None, 2, 54, 3, None],
                   "C": [20, 16, None, 3, 8],
                   "D": [14, 3, None, None, 6]})
print(df)
```

```
      A     B     C     D
0  12.0   NaN  20.0  14.0
1   4.0   2.0  16.0   3.0
2   5.0  54.0   NaN   NaN
3   NaN   3.0   3.0   NaN
4   1.0   NaN   8.0   6.0
```

[33]:
```
df.interpolate(method ='linear', limit_direction ='forward')
```

[33]:

	A	B	C	D
0	12.0	NaN	20.0	14.0
1	4.0	2.0	16.0	3.0
2	5.0	54.0	9.5	4.0
3	3.0	3.0	3.0	5.0
4	1.0	3.0	8.0	6.0

[35]:
```
dict = {'First Score': [100, 90, np.nan, 95],
        'Second Score': [30, np.nan, 45, 56],
        'Third Score': [52, 40, 80, 98],
        'Fourth Score': [np.nan, np.nan, np.nan, 65]}
df = pd.DataFrame(dict)
df
```

[35]:

	First Score	Second Score	Third Score	Fourth Score
0	100.0	30.0	52	NaN
1	90.0	NaN	40	NaN
2	NaN	45.0	80	NaN
3	95.0	56.0	98	65.0

[36]:
```
df.dropna() # Dropping rows with at least one null value
```

[36]:

	First Score	Second Score	Third Score	Fourth Score
3	95.0	56.0	98	65.0

[37]:
```
dict = {'First Score': [100, np.nan, np.nan, 95],
        'Second Score': [30, np.nan, 45, 56],
        'Third Score': [52, np.nan, 80, 98],
        'Fourth Score': [np.nan, np.nan, np.nan, 65]}
df = pd.DataFrame(dict)
```

[38]:
```
df
```

[38]:

	First Score	Second Score	Third Score	Fourth Score
0	100.0	30.0	52.0	NaN
1	NaN	NaN	NaN	NaN
2	NaN	45.0	80.0	NaN
3	95.0	56.0	98.0	65.0

[40]:
```
df.dropna(how='all') # Dropping rows with all null values
```

[40]:

	First Score	Second Score	Third Score	Fourth Score
0	100.0	30.0	52.0	NaN
2	NaN	45.0	80.0	NaN
3	95.0	56.0	98.0	65.0

[41]:
```
dict = {'First Score': [100, np.nan, np.nan, 95],
        'Second Score': [30, np.nan, 45, 56],
        'Third Score': [52, np.nan, 80, 98],
        'Fourth Score': [60, 67, 68, 65]}
df = pd.DataFrame(dict)
```

[42]:
```
df
```

[42]:

	First Score	Second Score	Third Score	Fourth Score
0	100.0	30.0	52.0	60
1	NaN	NaN	NaN	67
2	NaN	45.0	80.0	68
3	95.0	56.0	98.0	65

[43]:
```
df.dropna(axis=1) # Dropping columns with at least one null value
```

[43]:

	Fourth Score
0	60
1	67
2	68
3	65

[44]:
```
d = pd.read_csv(r"content\employees.csv")
nd = d.dropna(axis=0, how='any')

print("Old data frame length:", len(d))
print("New data frame length:", len(nd))
print("Rows with at least one missing value:", (len(d) - len(nd)))
```

```
Old data frame length: 1000
New data frame length: 764
Rows with at least one missing value: 236
```

[ ]:
```

```